CSE574 Introduction to Machine Learning

Adversarial Attack: An Overview

Jue Guo

University at Buffalo

April 1, 2024

Outline

1 What are adversarial attacks?

- The surprising findings by Szegedy (2013) and Goodfellow (2014)
- Example of attacks
- Physical Attacks

2 Basic Terminologies

- Defining attacks
- Multi-class Problem
- Three forms of attack
- Objective function and constraint sets

Why?

- Robustness = easiness to fail when input is perturbed. Perturbation can be in any kind. Robustness machine learning is a very rich topic.
- We will look at something very narrow, called **adversarial robusness**, also known as robustness against **attacks**.
- Adversairal attack is a very **hot** topic, as of today. We should not over-emphasize its importance. There are many other important problems.

Adversarial Attack Example: FGSM

- It is not difficult to fool a classifier
- The perturbation could be perceptually not noticeable

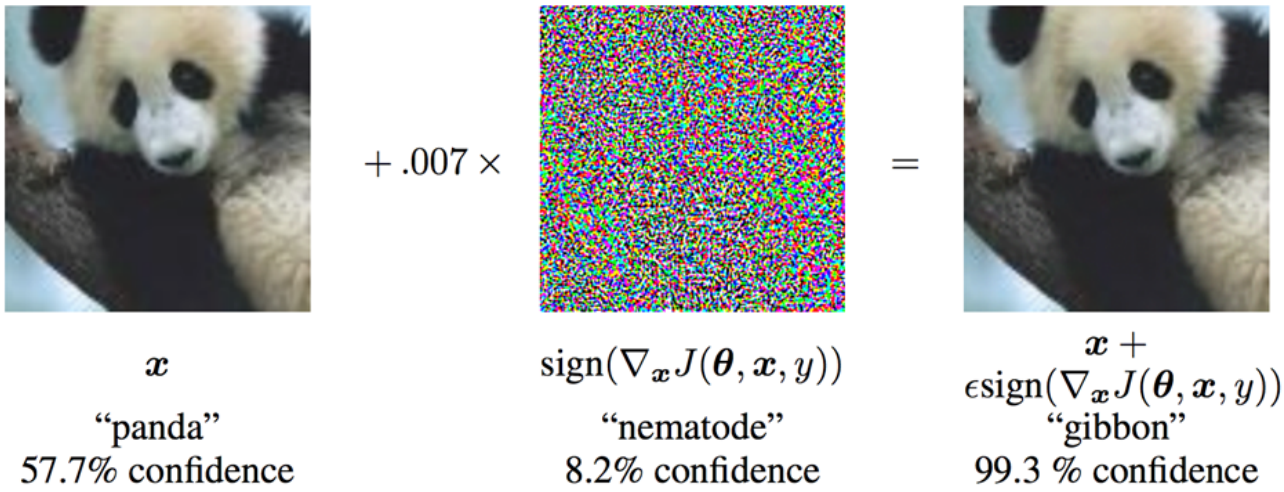


Figure: Goodfellow et al. "Explaining and Harnessing Adversarial Examples", https://arxiv.org/pdf/1412.6572.pdf

Adversarial Attack Example: Szegedy's 2013 Paper

- This paper actually appears one year before Goodfellow's 2014 paper.

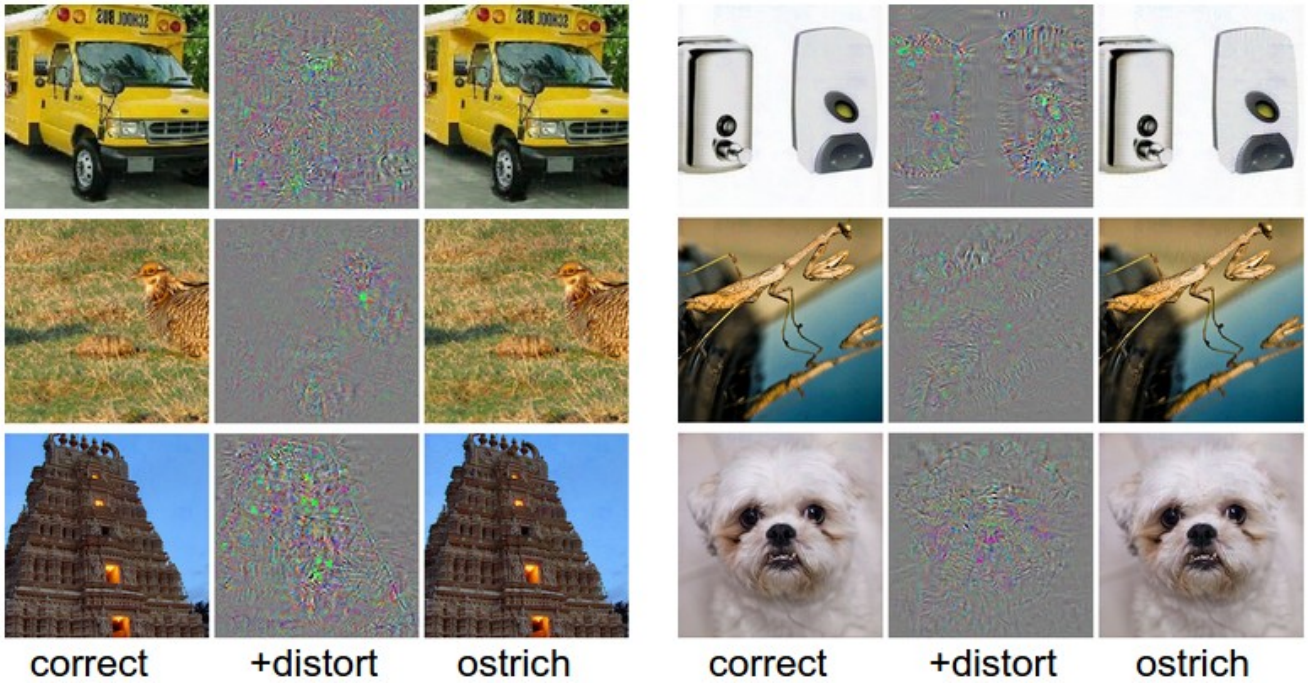


Figure: Szegedy et al. Intriguing properties of neural networks https://arxiv.org/abs/1312.6199

Adversarial Attack: Targeted Attack

- Targeted Attack

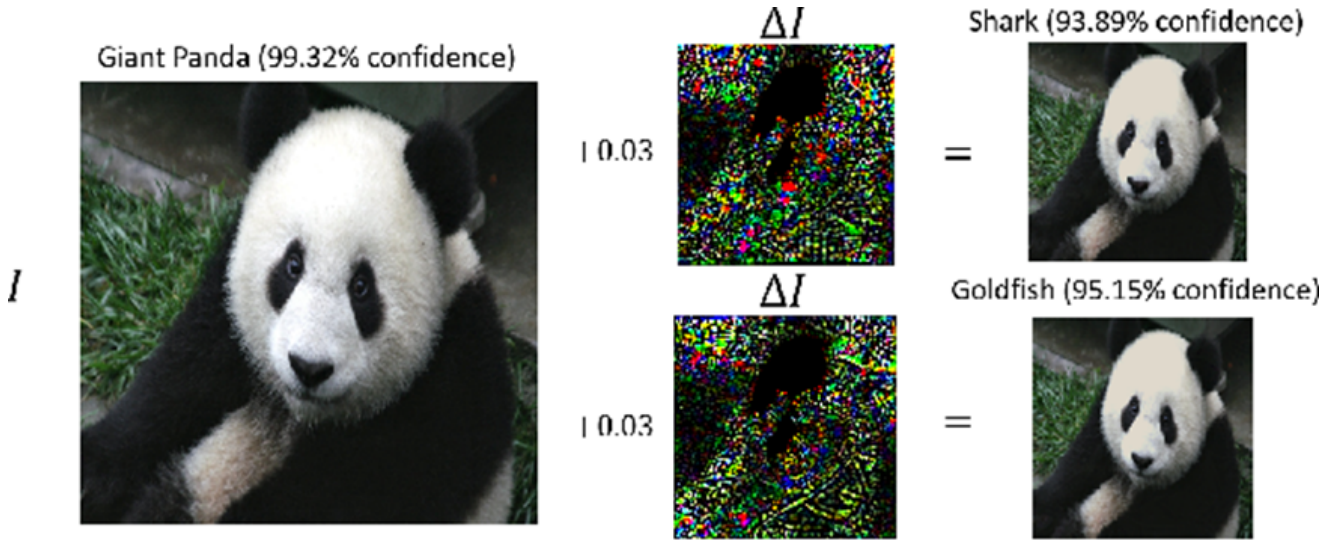


Figure: Adversarial Examples Detection in Deep Networks with Convolutional Filter Statistics, https://arxiv.org/abs/1612.07767

Adversarial Attack Example: One Pixel

- One-pixel Attack

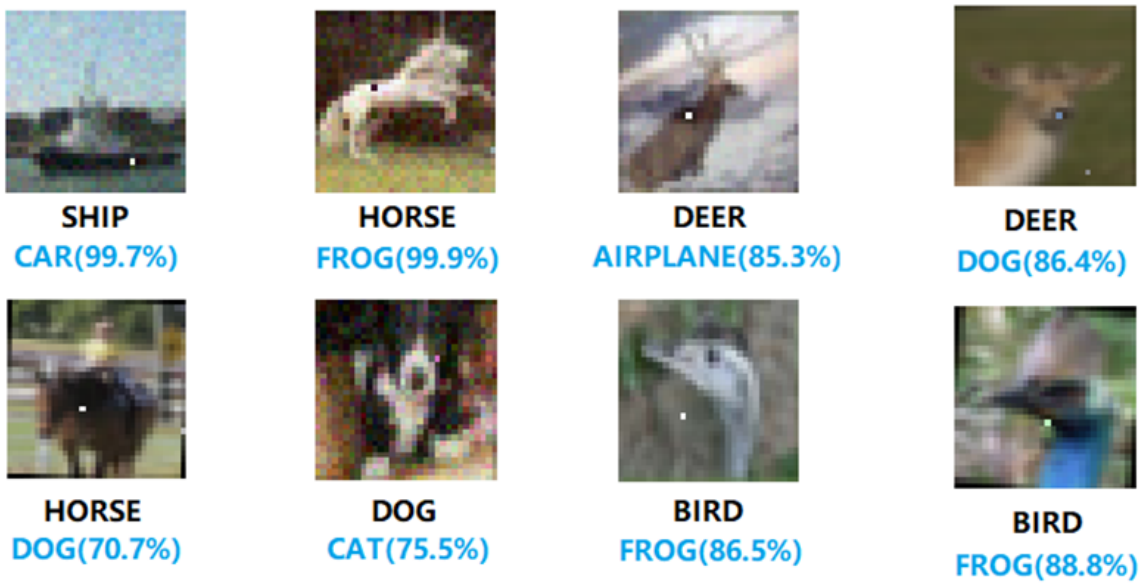


Figure: One pixel attack for fooling deep neural networks https://arxiv.org/abs/1710.08864

Adversarial Attack Example: Patch

- Adding a patch

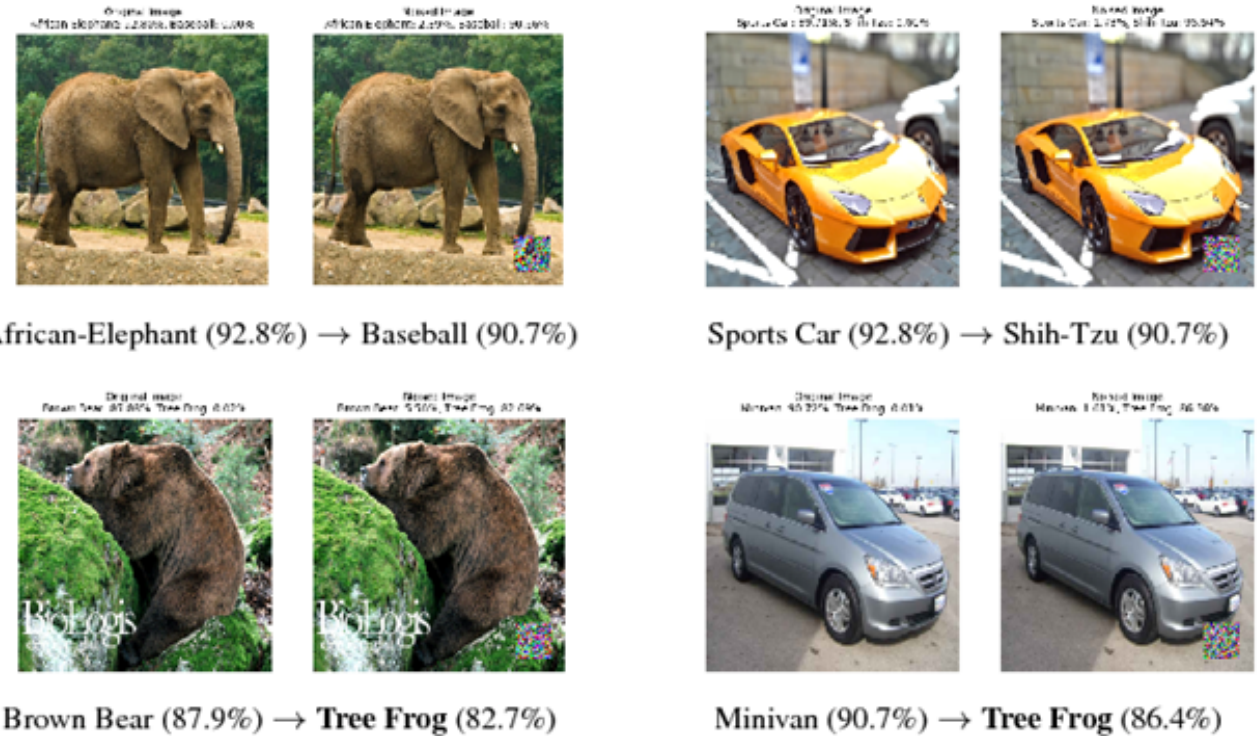


Figure: LaVAN: Localized and Visible Adversarial Noise, https://arxiv.org/abs/1801.02608

Adversarial Attack Example: Stop Sign

- The Michigan / Berkely Stop Sign

Figure: Robust Physical-World Attacks on Deep Learning Models
https://arxiv.org/abs/1707.08945

Adversarial Attack Example: Turtle

- The MIT 3D Turtle

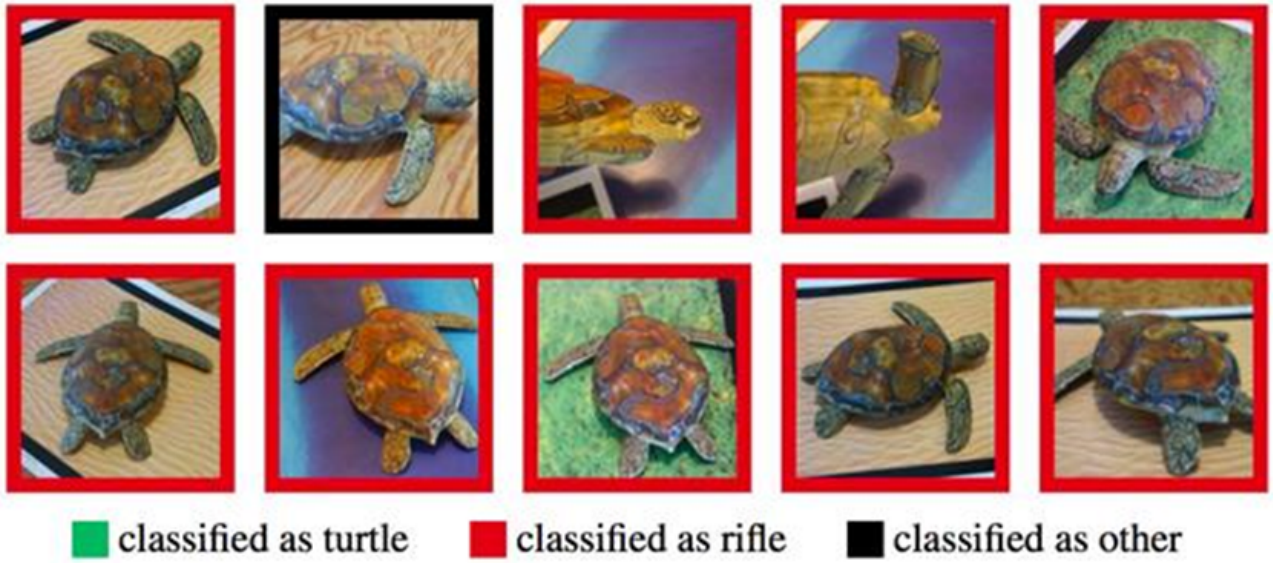


Figure: Synthesizing Robust Adversarial Examples https://arxiv.org/pdf/1707.07397.pdf
https://www.youtube.com/watch?v=YXy60X1iNoA

Adversarial Attack Example: Glass

- CMU Glass

Sharif, M., Bhagavatula, S., Bauer, L., & Reiter, M. K. (2016, October).
Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition.
In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (pp. 1528-1540). ACM.

Figure: Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition https://www.cs.cmu.edu/ ~sbhagava/papers/face-rec-ccs16.pdf https://www.archive.ece.cmu.edu/ ~lbauer/proj/advml.php

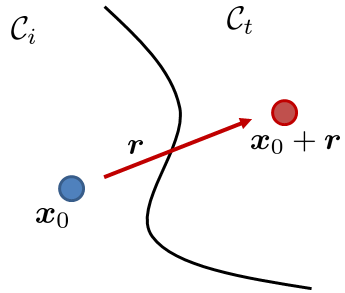
Definition: Additive Adversarial Attack

Additive Adversarial Attack

Let $x_0 \in \mathbb{R}^d$ be a data point belong to class $\mathcal{C}_i$. Define a target class $\mathcal{C}_t$ An additive adversarial attack is an addition of a perturbation $r \in \mathbb{R}^d$ such that the perturbed data

$$x = x_0 + r$$

is misclassified as $\mathcal{C}_t$.



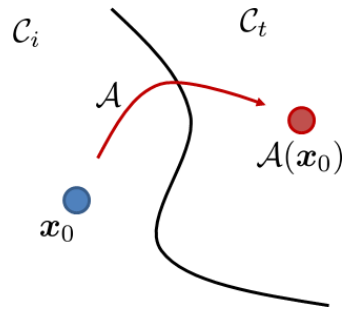
Definition: General Adversarial Attack

General Adversarial Attack

Let $x_0 \in \mathbb{R}^d$ be a data point belong to class $\mathscr{C}_i$. Define a target class $\mathscr{C}_t$ An adversarial attack is a mapping $\mathscr{A} : \mathbb{R}^d \to \mathbb{R}^d$ such that the perturbed data

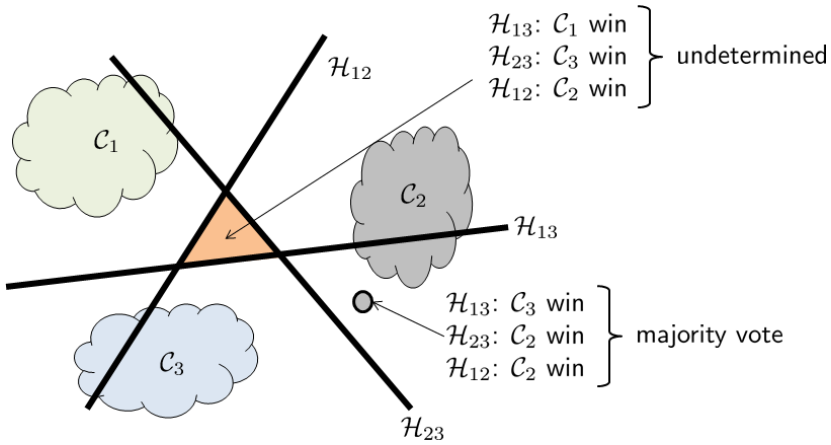
$$x = \mathscr{A}(x_0)$$

is misclassified as $\mathscr{C}_t$.



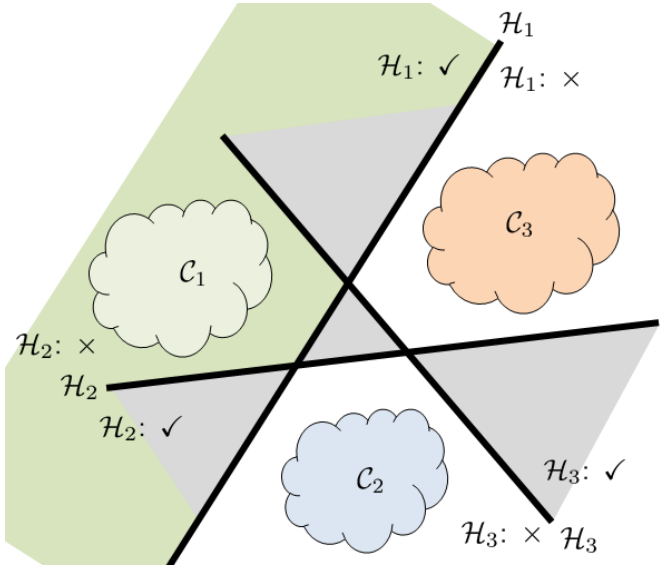
Multi-class Problem

Approach 1: One-on-One



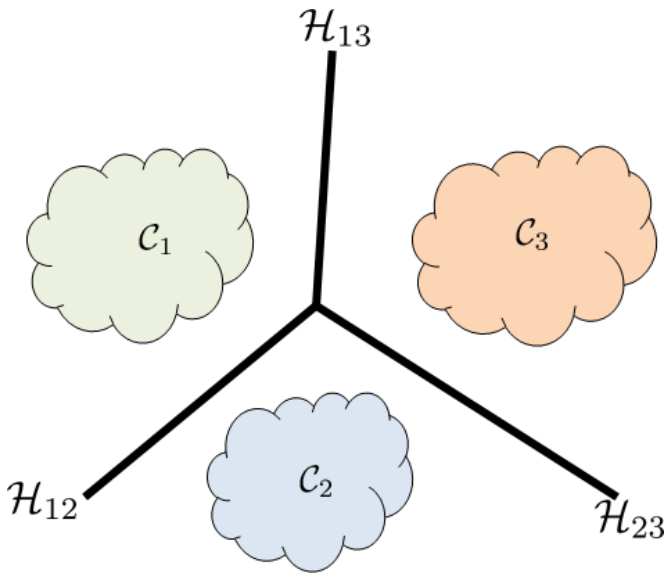
- Class i vs. Class j
- Give me a point, check which class has more votes
- There is an undetermined region

Approach 2: One-on-All



- Class i not Class i
- Give me a point, check which class has no conflict
- There are undetermined regions

Approach 3: Linear Machine



- Every point in the space gets assigned a class.
- You give me x, I compute $g_1(x), g_2(x), \ldots, g_K(x)$
- If $g_i(x) \geq g_j(x)$ for all $j \neq i$, then x belongs to class i

Correct Classification

- We are mostly interested in the linear machine problem.
- Let us try to simplify the notation. The statement:

$$\text{If } g_i(x) \geq g_j(x) \text{ for all } j \neq i, \text{ then } x \text{ belongs to class } i$$

is equivalent to (asking everyone to be less than 0)

$$\begin{aligned} g_1(x) - g_i(x) &\leq 0 \\ &\vdots \\ g_k(x) - g_i(x) &\leq 0 \end{aligned}$$

and is also equivalent to (asking the worst guy to be less than 0)

$$\max_{j \neq i} \{g_j(x)\} - g_i(x) \leq 0$$

- Therefore, if I want to launch an **adversarial attack**, I want to move you to class t:

$$\max_{j \neq t} \{g_j(x)\} - g_t(x) \leq 0$$

Our Approach

Here is what we are going to do

- First, we will preview the three **equivalent** forms of attack:
  - Minimum Distance Attack: Minimize the perturbation magnitude while accomplishing the attack objective.
  - Maximum Loss Attack: Maximize the training loss while ensuring perturbation is controlled.
  - Regularization-based Attack: Use regularization to control the amount of perturbation.
- Then, we will try to understand the **geometry** of the attacks.
- We will look at the **linear classifier** case to gain insights.

Minimum Distance Attack

Minimum Distance Attack

The **minimum distance attack** finds a perturbed data x by solving the optimization

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \|x - x_0\| \\ \text{subject to} \quad & \max_{j \neq t} \{g_j(x)\} - g_t(x) \leq 0 \end{aligned}$$

where $\|\cdot\|$ can be any norm specified by the user.

- I want to make you to class $\mathscr{C}_t$
- So the constraint needs to be satisfied.
- But I also want to minimize the attack strength. This gives the objective.

Maximum Loss Attack

Maximum Loss Attack

The **maximum loss attack** finds a perturbed data x by soling the optimization

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & g_t(x) - \max_{j \neq t} \{g_j(x)\} \\ \text{subject to} \quad & \|x - x_0\| \leq \eta \end{aligned}$$

where $\|\cdot\|$ can be any norm specified by the user, and $\eta > 0$ denotes the attack strength.

- I want to bound my attack $\|x - x_0\| \leq \eta$
- I want to make $g_t(x)$ as big as possible
- So I want to maximize $g_t(x) - \max_{j \neq t} \{g_j(x)\}$
- This is equivalent to

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \max_{j \neq t} \{g_j(x)\} - g_t(x) \\ \text{subject to} \quad & \|x - x_0\| \leq \eta \end{aligned}$$

Regularization-based Attack

Regularization-based Attack

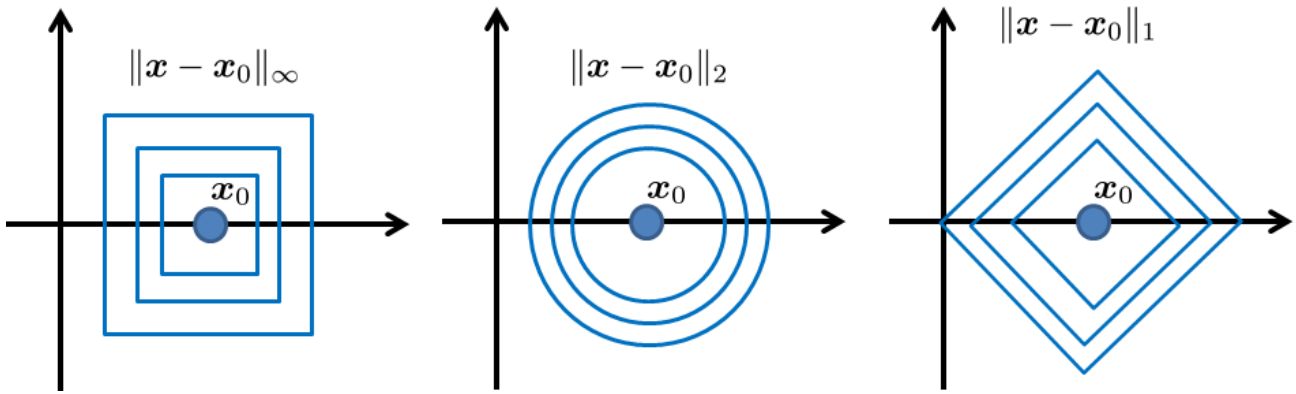
The **regularization-based attack** finds a perturbed data x by solving the optimization

$$\underset{x}{\text{minimize}} \|x - x_0\| + \lambda \left(\max_{j \neq t} \{g_j(x)\} - g_t(x) \right)$$

where $\|\cdot\|$ can be any norm specified by the user, and $\lambda > 0$ is a regularization parameter.

- Combine the two parts via regularization
- By adjusting $(\epsilon, \eta, \lambda)$, all three will give the same optimal value.

Understanding the Geometry: Objective Function



- ℓ_0-norm: $\varphi(x) = \|x - x_0\|_0$, which gives the most sparse solution. Useful when we want to limit the number of attack pixels.
- ℓ_1-norm: $\varphi(x) = \|x - x_0\|_1$, which is a convex surrogate of the ℓ_0.
- ℓ_∞-norm: $\varphi(x) = \|x - x_0\|_\infty$, which minimizes the maximum element of the perturbation.

Understanding the Geometry: Constraint

- The constraint set is

$$\Omega = \left\{ x \mid \max_{j \neq t} \{g_j(x)\} - g_t(x) \leq 0 \right\}$$

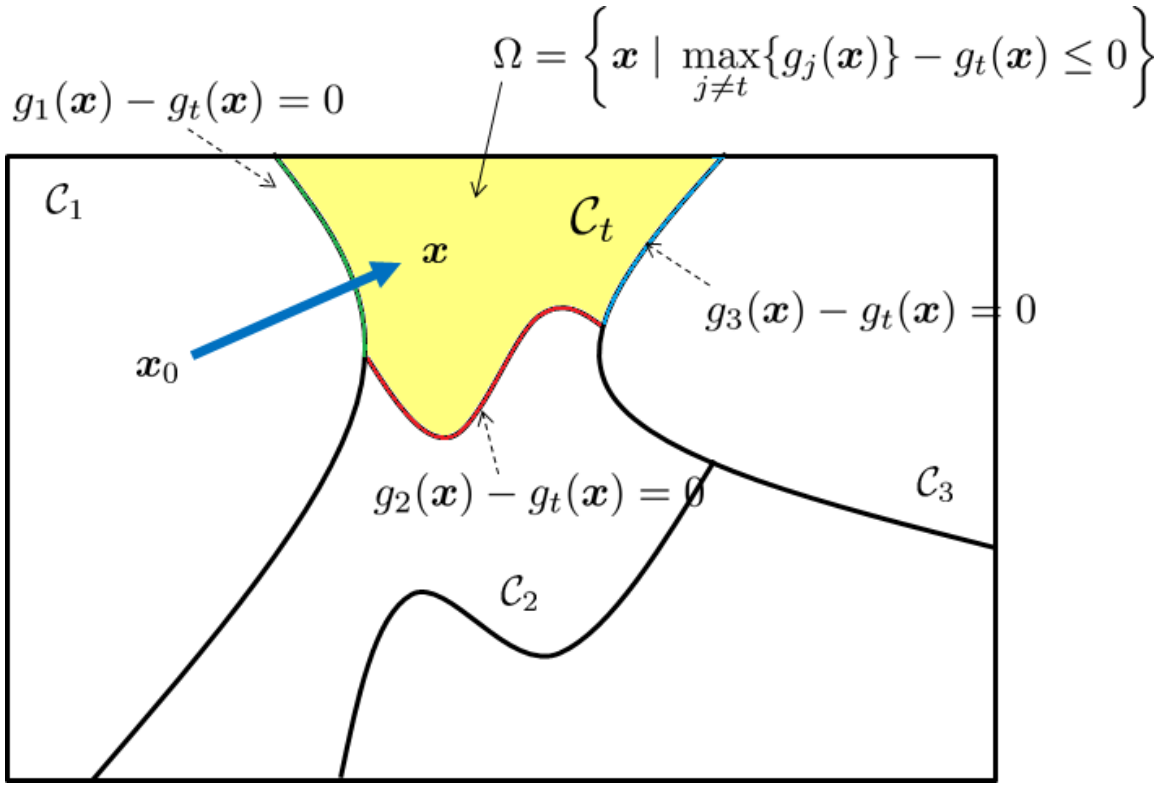
- We can write Ω as

$$\Omega = \left\{ x \;\middle|\; \begin{array}{c} g_1(x) - g_t(x) \leq 0 \\ g_2(x) - g_t(x) \leq 0 \\ \vdots \\ g_k(x) - g_t(x) \leq 0 \end{array} \right\}$$

- Remark: If you want to replace max by i^*, then i^* is a function of x:

$$\Omega = \{x \mid g_{i^*(x)}(x) - g_t(x) \leq 0\}$$

Understanding the Geometry: Constraint



Linear Classifier

- Let us take a closer look at the linear case.
  - Each discriminant function takes the form

$$g_i(x) = w_i^T x + w_{i,0}$$

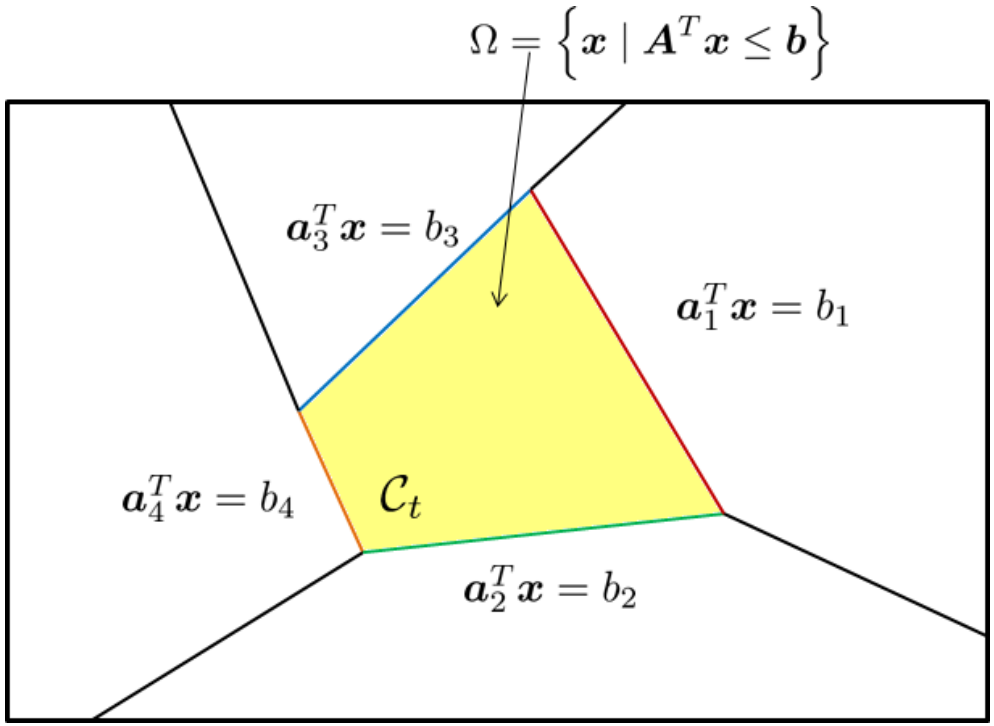
  - The decision boundary between the i-th class and the t-th class is therefore

$$g(x) = (w_i - w_t)^T x + w_{i,0} - w_{t,0} = 0$$

- The constraint set Ω is

$$\begin{bmatrix} w_1^T - w_t^T \\ \vdots \\ w_{t-1}^T - w_t^T \\ w_{t+1}^T - w_t^T \\ \vdots \\ w_k^T - w_t^T \end{bmatrix} x + \begin{bmatrix} w_{1,0} - w_{t,0} \\ \vdots \\ w_{t-1,0} - w_{t,0} \\ w_{t+1,0} - w_{t,0} \\ \vdots \\ w_{k,0} - w_{t,0} \end{bmatrix} \leq \mathbf{0} \Leftrightarrow A^T x \leq b$$

Linear Classifier



- You can show $\Omega = \{A^T x \leq b\}$ is convex.
- But the complement $\Omega^c = \{A^T x > b\}$ is not convex.
  - So targeted attack is easier to analyze than untargeted attack.

Attack: The Simplest Example

The optimization is:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \|x - x_0\| \\ \text{subject to} \quad & \max_{j \neq t} \{g_j(x)\} - g_t(x) \leq 0 \end{aligned}$$

- Suppose we use ℓ_2-norm, and consider **linear** classifiers, then the attack is given by

$$\underset{x}{\text{minimize}} \|x - x_0\|^2 \text{ subject to } A^T x \leq b$$

  - This is a **quadratic programming** problem.
  - We will discuss how to solve this problem analytically.

Summary

- Adversarial attack is a universal phenomenon for **any** classifier.
- Attacking deep networks are popular because people think that they are unbeatable.
- There is really nothing too magical behind adversarial attack.
  - All attacks are based on one of the forms of attacks.
  - Deep networks are trickier, as we will see, because the internal model information is not easy to extract.
  - We will learn the basic principles of attacks, and try to gain insights from linear models.